REMARKS

Pending claims

The pending claims are 2, 3, 5 and 8-11, of which claims 5, 9 and 10 are independent. Note that claim 5 is now the independent base claim for claims 2, 3, and 8, with claim 1 being canceled.

Rejections under 35 USC § 112

The Examiner rejected claims 1 and 5 under 35 USC § 112 because the limitation "the display widths" was indefinite. Claim 1 has been canceled in favor of claim 8, which no longer includes this term, and claim 5 has been amended to make clear that the display widths relate to the minimum widths necessary to display each of the various user-selectable choices. Claim 5 should therefore now be definite.

Rejections under 35 USC § 103

The examiner rejected claims 1-5 as being obvious in view of the admitted prior art and a hypothetical combination with Microsoft IE.

In particular, the Examiner wrote that the admitted prior art teaches "generating and displaying a graphical input device by executing a subroutine that is scripting embedded within the downloaded code."

The applicants believe that part of the problem is the notion of scripting being embedded in code. It is of course possible for developers at Microsoft Corp. (or some other browser developers) to embed code as a subroutine in Microsoft IE (or some other browser) that will do almost anything. The graphical display device that the browser generates for the user will then be different from the "expected" one, but will still be the result of actions remote from the user and the user's computer. Such a change would be "top down."

One distinction between this scenario and the applicants' invention as defined in claim 5 is that the subroutine or scripting is embedded and executed "in the **local** computer." One advantage of this is that the user's ability to benefit from the invention is not dependent on the browser's developers, or, indeed, on the particular browser at all, as long as the subroutine/scripting is in a language that can be executed along with the browser. Because of this independence and locality, without anything else, the user might be presented with conflicting graphical input devices, or with a graphical input device with which he is not familiar. Claim 5 therefore includes the further limitation of generating and displaying the graphical input device on the display screen "as a non-menu, text-input graphic device **in place of the defined drop-down menu** but having the general appearance of the drop-down menu." Both of these features are lacking in any prior art described in the applicants' specification, and especially in Microsoft IE.

The Examiner further wrote:

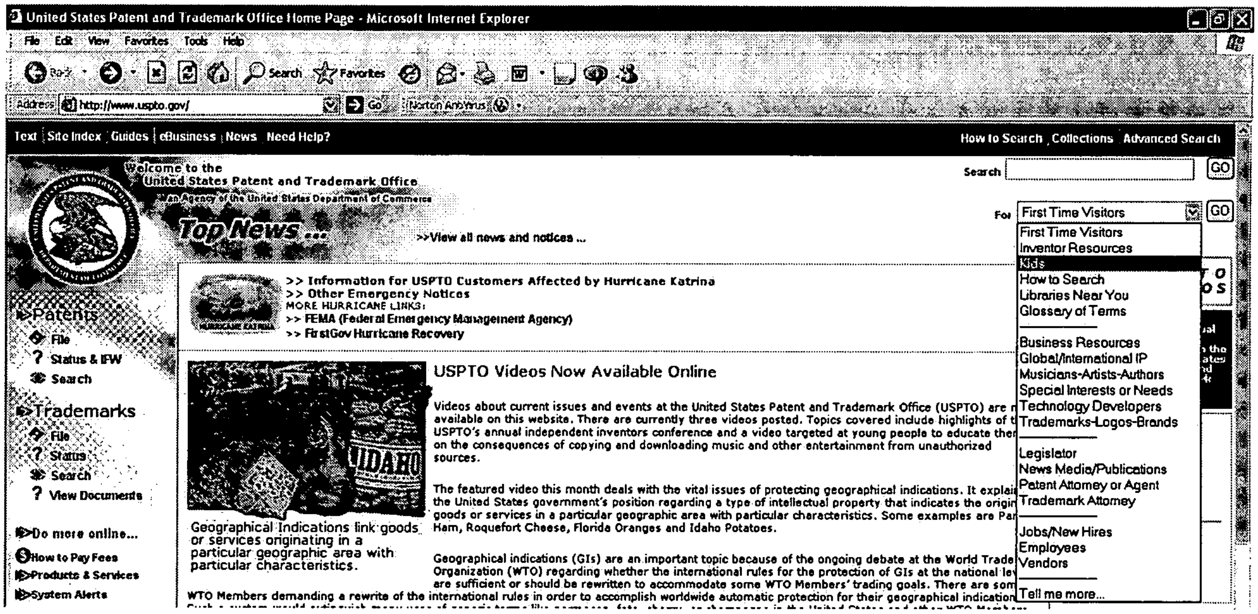
> The applicant's admitted prior art does not teach automatically choosing the second display width as a function of display widths of the user-dependent choices, such that the second display width is expandable relative to the first display width. Microsoft IE discloses a second display width (2 of fig. 2) is chosen as a function of display widths of the user-dependent choices, such that the second display width is expandable relative to the first display width (1 of fig. 2). It would have been obvious to an artisan at the time of the invention to apply the teaching from Microsoft IE to modify the applicant's prior art to have the second display width greater than the first display width since it would conserve the display space.

This relates to the "expandable display width" feature of the applicants' invention found in claim 5. The applicants believe that much of the rejection is due to misinterpretation of the term "expandable," much of which is in turn caused by it not being immediately clear just where the expansion is taking place, and what is doing it. The applicants are strengthened in this belief by the Examiner's assertion:

> In order for a system, software or application functions properly, it must be set up or configured for a first time. In this case, expanding the width of the pop-up window for a first time, then every time upon sensing user selection of the graphical input device, the pop-up window with expanded width relative to the first display width will be automatically displayed. Therefore, Microsoft IE does read on, the claim language of independent claims 1 and 5, especially the limitation, "automatically choosing the second display width as a function of the display widths of the user-dependent choices, such that the second display width is expandable relative to the first display width"

Of the prior art arrangements that generate any form of secondary display field (such the example in Microsoft IE given by the Examiner) are *static*, that is, the width with which they are initially generated is pre-determined and are either fixed or require user adjustment. In most cases, both the first and second display fields are fixed by whomever programs the HTML page and has access to the drop-down selections included or likely to be included later. There is a big difference between a system that automatically displays a window whose width has been pre-determined at the time of set-up and automatically choosing the width of the window locally: Manually programming in the width given the known or assumed possible entries is not "automatic."

The USPTO's own home page shows a clear example of this, in Microsoft IE:


United States Patent and Trademark Office Home Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://www.uspto.gov/ Go Norton AntiVirus
Text Site Index Guides eBusiness News Need Help?
How to Search Collections Advanced Search
Search GO
For First Time Visitors GO
First Time Visitors
Inventor Resources
Kids
How to Search
Libraries Near You
Glossary of Terms
Business Resources
Global/International IP
Musicians-Artists-Authors
Special Interests or Needs
Technology Developers
Trademarks-Logos-Brands
Legislator
News Media/Publications
Patent Attorney or Agent
Trademark Attorney
Jobs/New Hires
Employees
Vendors
Tell me more...
Welcome to the United States Patent and Trademark Office
An Agency of the United States Department of Commerce
Top News...
>>View all news and notices ...
Patents
File
Status & IFW
Search
Trademarks
File
Status
Search
View Documents
Do more online...
How to Pay Fees
Products & Services
System Alerts

>> Information for USPTO Customers Affected by Hurricane Katrina
>> Other Emergency Notices
MORE HURRICANE LINKS:
>> FEMA (Federal Emergency Management Agency)
>> FirstGov Hurricane Recovery

USPTO Videos Now Available Online

Videos about current issues and events at the United States Patent and Trademark Office (USPTO) are n available on this website. There are currently three videos posted. Topics covered include highlights of t USPTO's annual independent inventors conference and a video targeted at young people to educate ther on the consequences of copying and downloading music and other entertainment from unauthorized sources.

The featured video this month deals with the vital issues of protecting geographical indications. It explai the United States government's position regarding a type of intellectual property that indicates the origin goods or services in a particular geographic area with particular characteristics. Some examples are Pa Ham, Roquefort Cheese, Florida Oranges and Idaho Potatoes.

Geographical Indications link goods or services originating in a particular geographic area with particular characteristics.

Geographical indications (GIs) are an important topic because of the ongoing debate at the World Trade Organization (WTO) regarding whether the international rules for the protection of GIs at the national lev are sufficient or should be rewritten to accommodate some WTO Members' trading goals. There are som WTO Members demanding a rewrite of the international rules in order to accomplish worldwide automatic protection for their geographical indication

There is no way for the user even to adjust the width of the "For" field (an initial display field) or the field showing the drop-down choices. Moreover, the drop-down display field is at least three characters wider than it needs to be to display the choice "Trademarks-Logos-Brands" with the greatest character width.

The other known case is that where the first or second display widths can be adjusted by some user action. For example, as the applicants showed in their response to the previous Office action – using the same web site the Examiner used – the Address field of the Address Bar in Microsoft IE's address field can be unlocked and *manually* adjusted (dragged) by the user. The second display width was also *user adjustable* by dragging a resizing tab, but the user can drag it so as to make the second display width *contract* (become narrower) as well as expand (become wider) relative to the initial display field; consequently, the user can go too far in "both directions," either cutting off entries, or making the second field so wide that it has unnecessary right-aligned blank spaces. Moreover, whether the drop-down list (the second display field) was wider or narrower than the initial field was purely a matter of chance, or rather, how the user happened to have set it by adjusting the field on the toolbar.

In short, the prior art has one or both of two disadvantages for users, and sometimes both: either users must accept width decisions (including "conservative" estimates that leave lots of room for future, wider input choices) pre-determined by whomever wrote the HTML code, or they require the user to manually resize one or both of the initial display field or the secondary, drop-down field. This could possibly require the user to make several adjustments if an entry far down in a drop-down list happens to be even wider than a wide one higher up: The user might resize the window to show all of the higher-up entry without realizing that an even wider choice will appear when he scrolls further down.

Claim 5 has been amended to make it even clearer just how the invention goes about generating the second input field so as to make it both flexible, space-efficient and transparent to the user (important differences relative to the cited prior art are emphasized)

in the subroutine, ***locally*** **determining a greatest one of the minimum required display widths**;

sensing user selection of the graphical input device;

upon sensing user selection of the graphical input device, displaying on the screen a list of the user-selectable choices, the list having **a second display width equal to the greatest minimum required display widths;**

displaying at least a portion of the selected user-selectable choice in the data entry field and setting the input parameter to the selected user-selectable choice; and

locally, automatically and **dynamically** choosing the second display width as a function of the minimum required display widths of the user-selectable choices, such that the second display width is automatically and dynamically expanded relative to the first display width;

Compared with the prior art that the Examiner has cited (that found in the applicants' own specification and in Microsoft's products Outlook and IE) this aspect of the invention provides several advantages to a user:

- code- and vendor-independence – he does not have to wait for a remote vendor to adjust the downloaded code, or to choose to program in more convenient widths for display fields
- the initially displayed field may be kept compact
- the drop-down display field will not truncate entries, but rather will be wide enough to provide a full view of even the widest choice without being "too wide"
- flexibility – the width of drop-down field will be changed locally if the widths of the downloaded choices change
- ***dynamic*** (as opposed to static) and ***automatic*** width adjustment – again, the width of the drop-down field does not have to be pre-determined by the original programmer, and will adjust without manual intervention
- transparency – the user does not need to do anything to enjoy these benefits since they're all provided automatically

Claim 5 therefore defines features not found in any of the cited prior art – whether viewed separately or in combination – that provide clear and to date unavailable advantages to the user. As such, the invention as defined in this claim is both novel and non-obvious, such that claim 5 and its dependent claims should be allowed.

New independent claim 9 relates to two features: 1) expanded matching, whereby the system matches a concatenated *sequence* of entered characters character-by-character, as opposed to matching only on individual key entries without regard to previous entries; and 2) multi-field searching, that is, the feature of the invention originally found in now canceled claim 7, namely, sequential, character-by-character of searching for matches in more than one of a set of delimited fields of the user-selectable input choices.

As for expanded matching, the problem with the prior art is discussed in the specification in paragraphs [0026]-[0028], and the manner in which the invention solves this problem is discussed in paragraphs [0062]-[0064]. The applicants have also already explained both the problem and the invention's solution in their previous Office action response.

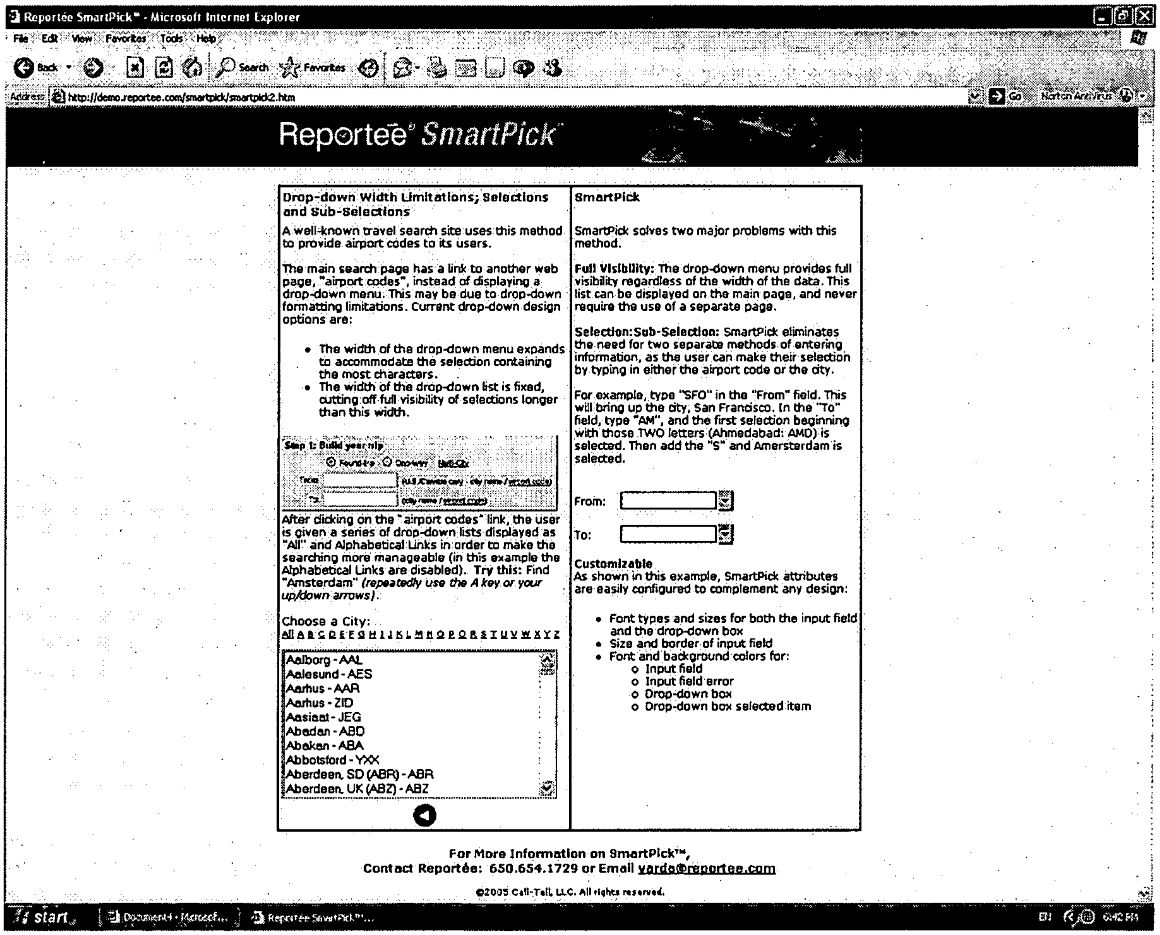
The concept of multi-field searching and matching was, as mentioned, found in previous claim 7, and is explained in specification paragraphs [0065], [0066], and part of [0068] ("The ability of the invention to perform intelligent matches that **start** with the lowest sub-items (to the right of the right-most delimiter) in selection lists arranged hierarchically not only reduces how much typing the user must do, but it also enables easy navigation of long lists" -- emphasis added). The example given in the specification (with reference to Figure 2B) is that a user could reach the entry "CODE 05:Upgrade" either conventionally, by typing "C-O-D-E-_-0-5" or, as preferred in the invention, simply by typing "U" (since no other right-field entries begin with the letter U.)

The examiner rejected the previous claim to this feature as being obvious in view of Microsoft IE and Official Notice. In particular, the Examiner asserted that searching to the right of a delimiter would be an obvious design choice. This rejection is understandable, since searching "beginning to right of a delimiting character" was the main point of novelty included in previous claim 7. New claim 9, however, includes additional limitations to better define the multi-field searching feature of the invention, especially lines 18-23:

> if no initial information field of the user-selectable choices has a character sequence matching the user-entered character sequence, highlighting for the user at least one user-selectable choice in a different information field whose initial characters match the user-entered character sequence, whereby a matching user-selectable choice can be located in more than one information field based on character-by-character comparison with the same user-entered character sequence

Note that such a feature is not found in any aspect of Microsoft IE. Nor is it found in, for example, file-locating routines of operating systems such as Microsoft Windows. Rather, these systems also treat hierarchical names as single text strings that are searched for the occurrence of input characters all at once, not sequentially, character-by-character. In particular, this feature is not found in the prior art with respect to any method of accessing entries in a graphical input device that resembles a pull-down menu.

To better understand the advantages – including multi-field searching -- of the invention, the applicants present here a series of screenshots that show an implementation of the applicants' invention used for purpose of demonstration. In this demonstrative implementation, what appears, to the user, to be a standard drop-down menu but is in reality the graphical input device according to the invention. In the implementation, over 3000 entries are available via the drop-down list when a user selects ("clicks on") the initial display field – this large number makes scrolling particularly cumbersome.


Reportée SmartPick™ - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://demo.reportee.com/smartpick/smartpick2.htm
Go Norton AntiVirus
Reportée® SmartPick™
Step 1: Build your trip
start Document4 - Microsoft... Reportée SmartPick™...

Drop-down Width Limitations; Selections and Sub-Selections

A well-known travel search site uses this method to provide airport codes to its users.

The main search page has a link to another web page, "airport codes", instead of displaying a drop-down menu. This may be due to drop-down formatting limitations. Current drop-down design options are:

- The width of the drop-down menu expands to accommodate the selection containing the most characters.
- The width of the drop-down list is fixed, cutting off full visibility of selections longer than this width.

After clicking on the "airport codes" link, the user is given a series of drop-down lists displayed as "All" and Alphabetical Links in order to make the searching more manageable (in this example the Alphabetical Links are disabled). Try this: Find "Amsterdam" *(repeatedly use the A key or your up/down arrows)*.

Choose a City:
All A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Aalborg - AAL
Aalesund - AES
Aarhus - AAR
Aarhus - ZID
Aasiaat - JEG
Abadan - ABD
Abakan - ABA
Abbotsford - YXX
Aberdeen, SD (ABR) - ABR
Aberdeen, UK (ABZ) - ABZ

SmartPick

SmartPick solves two major problems with this method.

Full Visibility: The drop-down menu provides full visibility regardless of the width of the data. This list can be displayed on the main page, and never require the use of a separate page.

Selection:Sub-Selection: SmartPick eliminates the need for two separate methods of entering information, as the user can make their selection by typing in either the airport code or the city.

For example, type "SFO" in the "From" field. This will bring up the city, San Francisco. In the "To" field, type "AM", and the first selection beginning with those TWO letters (Ahmedabad: AMD) is selected. Then add the "S" and Amersterdam is selected.

From:

To:

Customizable
As shown in this example, SmartPick attributes are easily configured to complement any design:

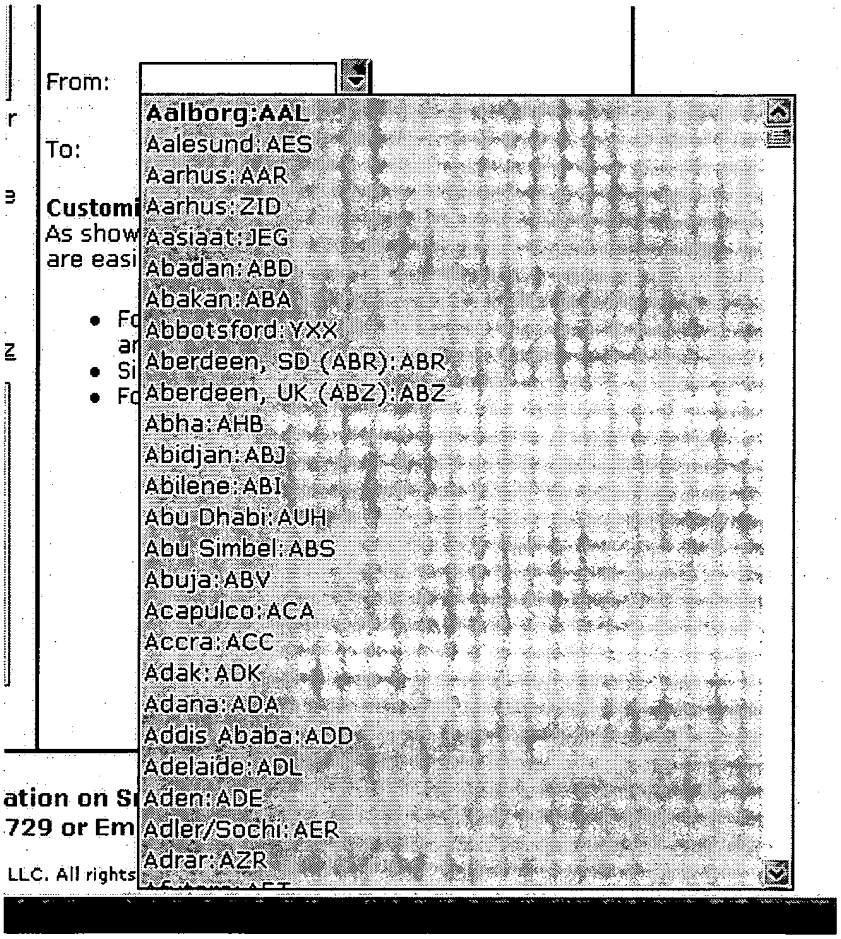
- Font types and sizes for both the input field and the drop-down box
- Size and border of input field
- Font and background colors for:
  - Input field
  - Input field error
  - Drop-down box
  - Drop-down box selected item

For More Information on SmartPick™,
Contact Reportée: 650.654.1729 or Email yardo@reportee.com

©2005 Call-Tell, LLC. All rights reserved.

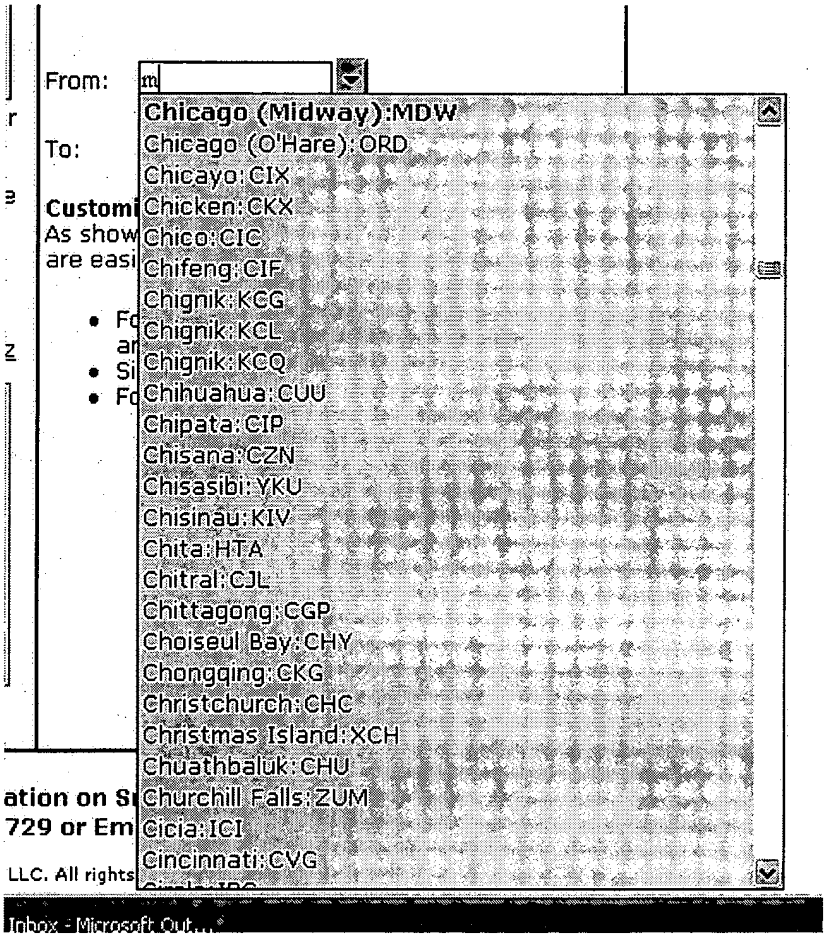
Screen shot 1

Assume the user wants to select the airport she will be traveling From. She will then click in the From information input field, which invokes the claim element "sensing user selection of the graphical input device." Then, again as in claim 5, "upon sensing user selection of the graphical input device, [the system displays] on the screen a list of the user-dependent selectable choices, the list having a second display width equal to the greatest minimum required display widths." See Screen shot 2:



Screen shot 2

Assume case-insensitivity and that the user wishes to select "Munich." Naturally, the user begins by typing "m." As claim 9 recites, the invention then begins "sequentially and character-by-character searching of at least an initial information field for each of the user-selectable choices according to each user-entered character; for each character sequence of at least one user-entered character that matches a corresponding character sequence in the initial information field of at least one user-selectable choice, highlighting at least one of the matching choices for the user." The <International Airport Code> field is the initial field, so the invention finds and highlights an entry whose code begins with m:



From: m

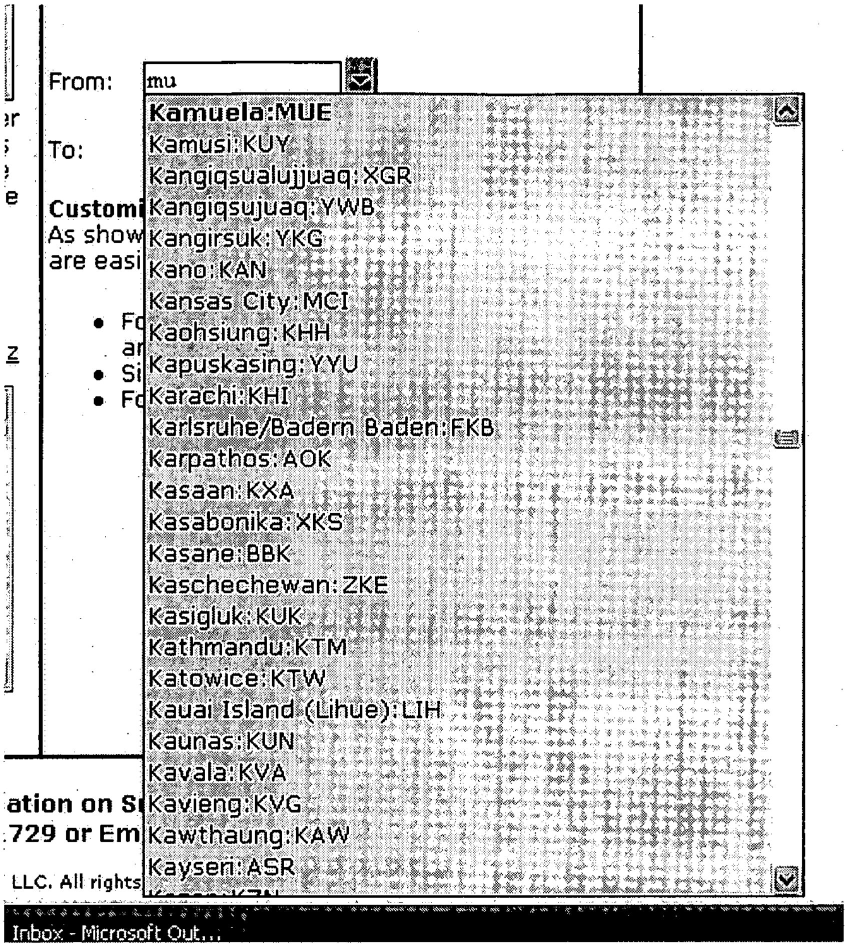
To:

Chicago (Midway):MDW
Chicago (O'Hare):ORD
Chicayo:CIX
Chicken:CKX
Chico:CIC
Chifeng:CIF
Chignik:KCG
Chignik:KCL
Chignik:KCQ
Chihuahua:CUU
Chipata:CIP
Chisana:CZN
Chisasibi:YKU
Chisinau:KIV
Chita:HTA
Chitral:CJL
Chittagong:CGP
Choiseul Bay:CHY
Chongqing:CKG
Christchurch:CHC
Christmas Island:XCH
Chuathbaluk:CHU
Churchill Falls:ZUM
Cicia:ICI
Cincinnati:CVG

Screen shot 3

Note that this yields an airport whose name does not start with m, but rather Chicago.

The user will next type a "u", which leads to Screen shot 4:



From: mu

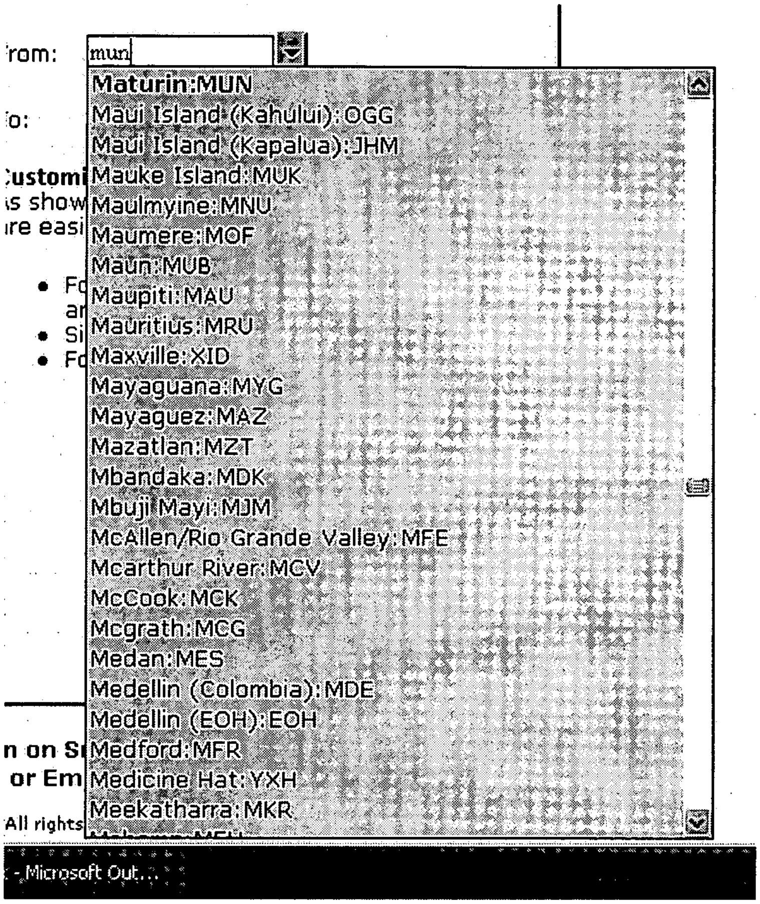
To:

Kamuela:MUE
Kamusi:KUY
Kangiqsualujjuaq:XGR
Kangiqsujuaq:YWB
Kangirsuk:YKG
Kano:KAN
Kansas City:MCI
Kaohsiung:KHH
Kapuskasing:YYU
Karachi:KHI
Karlsruhe/Badern Baden:FKB
Karpathos:AOK
Kasaan:KXA
Kasabonika:XKS
Kasane:BBK
Kaschechewan:ZKE
Kasigluk:KUK
Kathmandu:KTM
Katowice:KTW
Kauai Island (Lihue):LIH
Kaunas:KUN
Kavala:KVA
Kavieng:KVG
Kawthaung:KAW
Kayseri:ASR

Screen shot 4

The invention thus highlights the airport Kamuela (Hawaii), whose *code* begins with "mu". (Note that this also demonstrates the advantage of extended matching, that is, not searching u as an independent character, but rather as a continuation of the character sequence that began with m.)

The user will next type a "n", which, together with the previous "mu" leads to Screen shot 5:



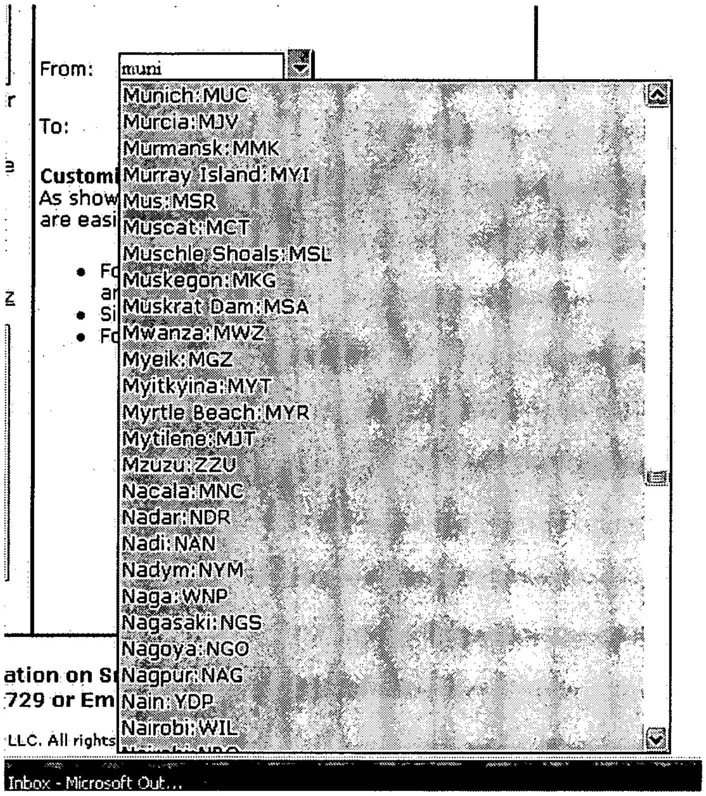
mun

Maturin:MUN
Maui Island (Kahului):OGG
Maui Island (Kapalua):JHM
Mauke Island:MUK
Maulmyine:MNU
Maumere:MOF
Maun:MUB
Maupiti:MAU
Mauritius:MRU
Maxville:XID
Mayaguana:MYG
Mayaguez:MAZ
Mazatlan:MZT
Mbandaka:MDK
Mbuji Mayi:MJM
McAllen/Rio Grande Valley:MFE
Mcarthur River:MCV
McCook:MCK
Mcgrath:MCG
Medan:MES
Medellin (Colombia):MDE
Medellin (EOH):EOH
Medford:MFR
Medicine Hat:YXH
Meekatharra:MKR

Screen shot 5

Thus, "mun" gives Maturin (Venezuela).

This is still not what the user wants, however, so she continues entering characters by typing "i".



From: muni

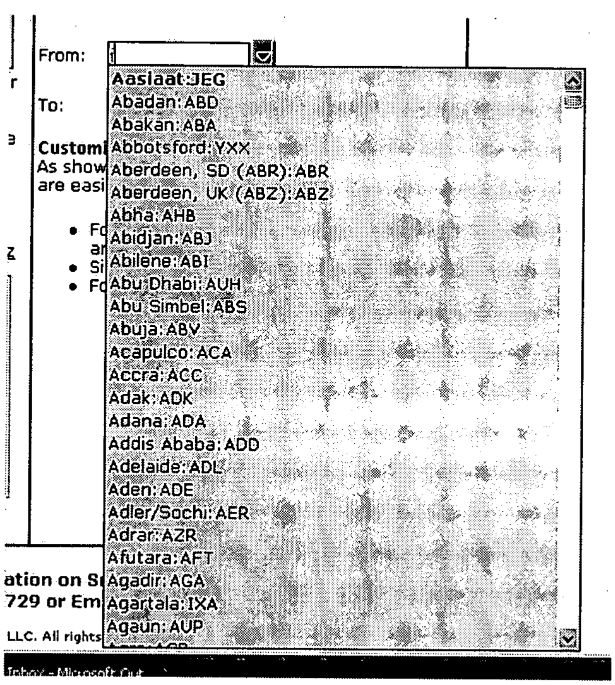
To:

Munich:MUC
Murcia:MJV
Murmansk:MMK
Murray Island:MYI
Mus:MSR
Muscat:MCT
Muschle Shoals:MSL
Muskegon:MKG
Muskrat Dam:MSA
Mwanza:MWZ
Myeik:MGZ
Myitkyina:MYT
Myrtle Beach:MYR
Mytilene:MJT
Mzuzu:ZZU
Nacala:MNC
Nadar:NDR
Nadi:NAN
Nadym:NYM
Naga:WNP
Nagasaki:NGS
Nagoya:NGO
Nagpur:NAG
Nain:YDP
Nairobi:WIL

Screen shot 6

Now, Munich appears at the top as the current selection and the user may select it in any conventional manner. This example illustrates how the invention searches sequentially in the code field, but even locates Munich in the name field if no code matches. In this case, failure to match was simply because of entry of a fourth character "i" and no airport code has a fourth character at all. The user does not have to tell the invention to look in the name field since this is an automatic result of the invention's search feature as defined in claim 9.

Screen shots 7-9 illustrate that the invention's multi-field search feature (claim 9: "if no initial information field of the user-selectable choices has a character sequence matching the user-entered character sequence, highlighting for the user at least one user-selectable choice in a different information field whose initial characters match the user-entered character sequence") can lead to a match being found in a different information field based on fewer than four characters (that is, no greater than the number of characters in the widest entry in the initial information field). Here, assume that a traveler wants to leave from Juneau (Alaska). Typing j-u-n gives, progressively, Aasiaat:JEG (Screen shot 7) in Greenland, Juiz De Fora:JDF (Screen shot 8) in Brazil, and, finally, Juneau:JNU.

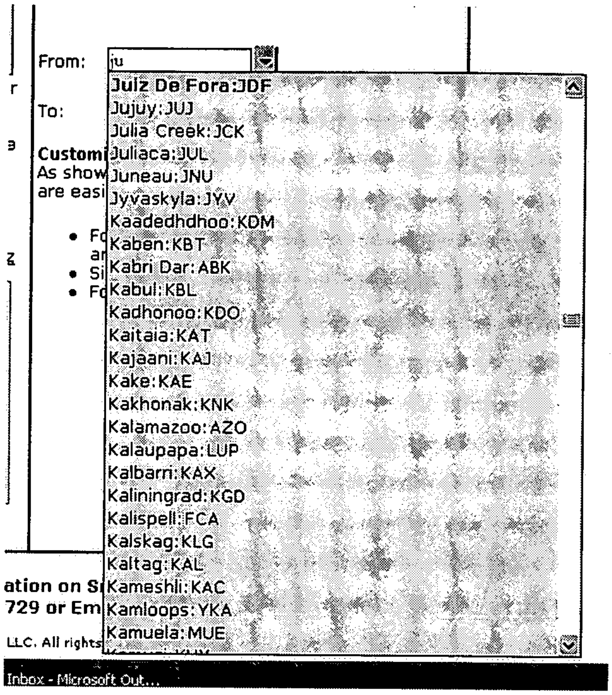


From: j

To:

Aasiaat:JEG
Abadan:ABD
Abakan:ABA
Abbotsford:YXX
Aberdeen, SD (ABR):ABR
Aberdeen, UK (ABZ):ABZ
Abha:AHB
Abidjan:ABJ
Abilene:ABI
Abu Dhabi:AUH
Abu Simbel:ABS
Abuja:ABV
Acapulco:ACA
Accra:ACC
Adak:ADK
Adana:ADA
Addis Ababa:ADD
Adelaide:ADL
Aden:ADE
Adler/Sochi:AER
Adrar:AZR
Afutara:AFT
Agadir:AGA
Agartala:IXA
Agaun:AUP

Screen shot 7

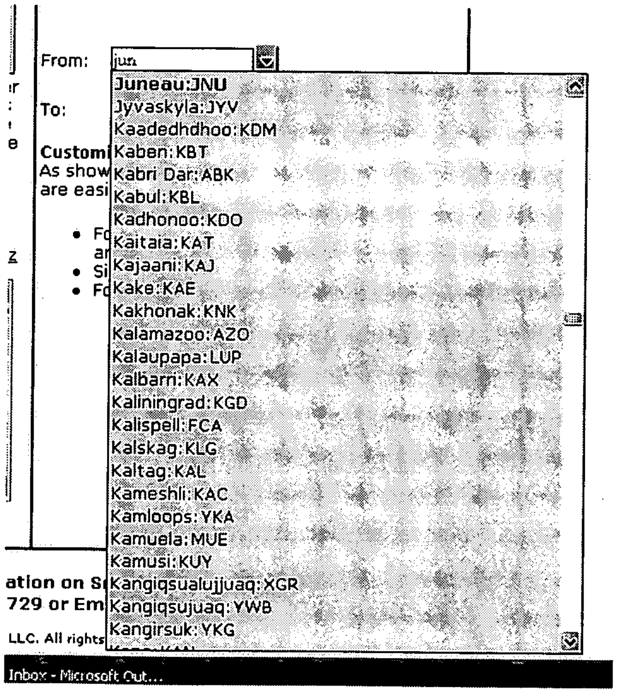


From: ju

To:

Juiz De Fora:JDF
Jujuy:JUJ
Julia Creek:JCK
Juliaca:JUL
Juneau:JNU
Jyvaskyla:JYV
Kaadedhdhoo:KDM
Kaben:KBT
Kabri Dar:ABK
Kabul:KBL
Kadhonoo:KDO
Kaitaia:KAT
Kajaani:KAJ
Kake:KAE
Kakhonak:KNK
Kalamazoo:AZO
Kalaupapa:LUP
Kalbarri:KAX
Kaliningrad:KGD
Kalispell:FCA
Kalskag:KLG
Kaltag:KAL
Kameshli:KAC
Kamloops:YKA
Kamuela:MUE

Screen shot 8



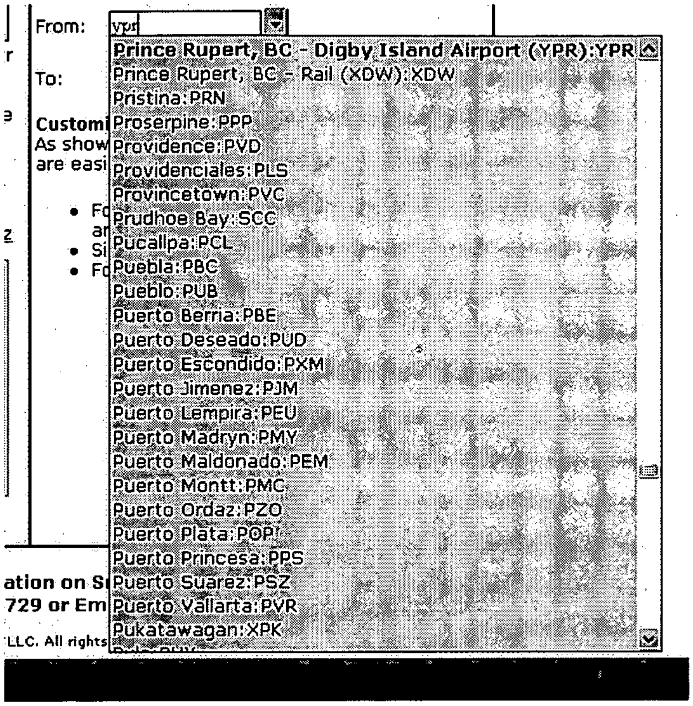
From: jun

To:

Juneau:JNU
Jyvaskyla:JYV
Kaadedhdhoo:KDM
Kaben:KBT
Kabri Dar:ABK
Kabul:KBL
Kadhonoo:KDO
Kaitaia:KAT
Kajaani:KAJ
Kake:KAE
Kakhonak:KNK
Kalamazoo:AZO
Kalaupapa:LUP
Kalbarri:KAX
Kaliningrad:KGD
Kalispell:FCA
Kalskag:KLG
Kaltag:KAL
Kameshli:KAC
Kamloops:YKA
Kamuela:MUE
Kamusi:KUY
Kangiqsualujjuaq:XGR
Kangiqsujuaq:YWB
Kangirsuk:YKG

Screen shot 9

Now assume the traveler wants to select the airport in Prince Rupert, British Columbia, Canada. The full name-field portion of the entry for this airport is "Prince Rupert, BC – Digby Island Airport (YPR)". The invention provides two ways to reach this entry, although the user does not need to know this or learn any special data entry procedure: enter the airport code "ypr", as in Screen shot 10:


From: ypr

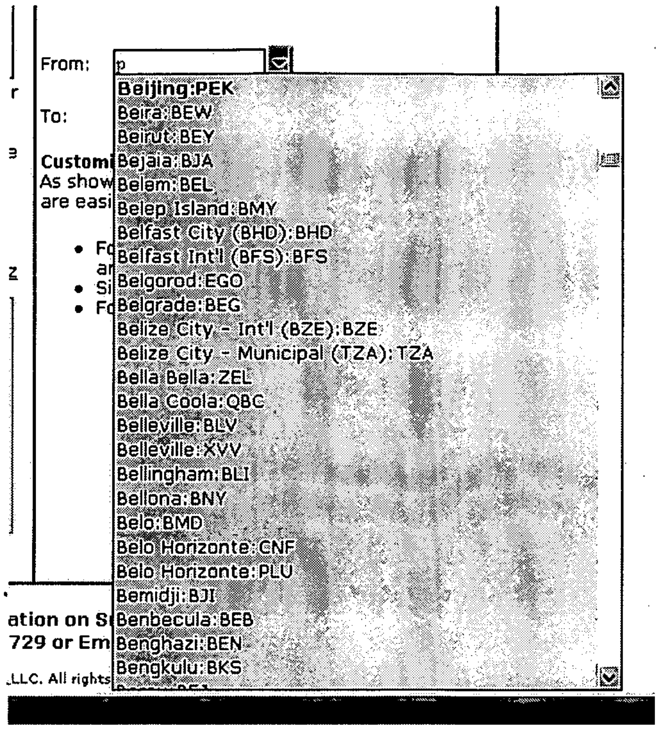
To:

Prince Rupert, BC - Digby Island Airport (YPR):YPR
Prince Rupert, BC - Rail (XDW):XDW
Pristina:PRN
Proserpine:PPP
Providence:PVD
Providenciales:PLS
Provincetown:PVC
Prudhoe Bay:SCC
Pucallpa:PCL
Puebla:PBC
Pueblo:PUB
Puerto Berria:PBE
Puerto Deseado:PUD
Puerto Escondido:PXM
Puerto Jimenez:PJM
Puerto Lempira:PEU
Puerto Madryn:PMY
Puerto Maldonado:PEM
Puerto Montt:PMC
Puerto Ordaz:PZO
Puerto Plata:POP
Puerto Princesa:PPS
Puerto Suarez:PSZ
Puerto Vallarta:PVR
Pukatawagan:XPK

Screen shot 10

or begin to type the name of the airport, as shown in Screen shots 11-17, which show the following results for each partial entry:

Screen shot	Entry sequence	Result
11	p	Beijng:PEK
12	pr	Parana:PRA
13	pri	Prince Albert:YPA
14	prin	Prince Albert:YPA
15	princ	Prince Albert:YPA
16	prince	Prince Albert:YPA
17	prince r	Prince Rupert, BC – Digby Island Airport (YPR):YPR



From: p

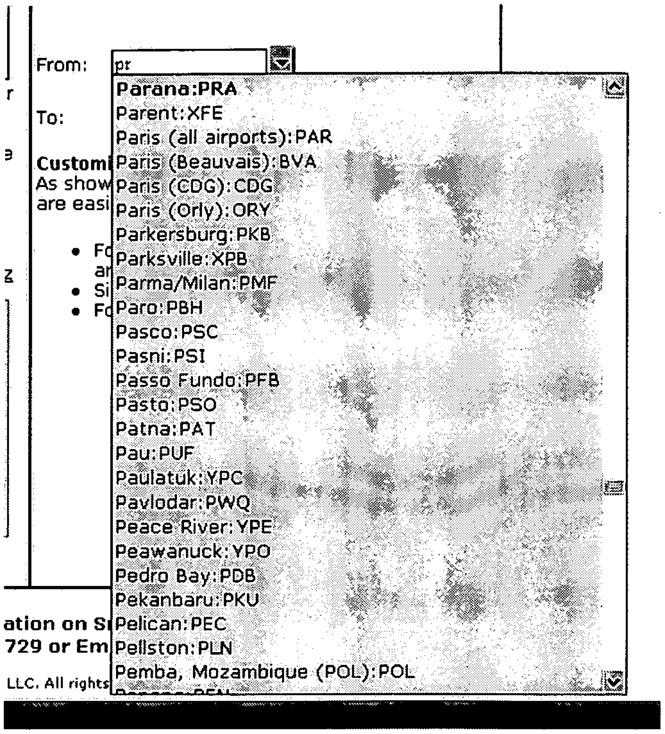
To:

Customi
As show
are easi

ation on S
729 or Em
LLC. All rights

Beijing:PEK
Beira:BEW
Beirut:BEY
Bejaia:BJA
Belem:BEL
Belep Island:BMY
Belfast City (BHD):BHD
Belfast Int'l (BFS):BFS
Belgorod:EGO
Belgrade:BEG
Belize City - Int'l (BZE):BZE
Belize City - Municipal (TZA):TZA
Bella Bella:ZEL
Bella Coola:QBC
Belleville:BLV
Belleville:XVV
Bellingham:BLI
Bellona:BNY
Belo:BMD
Belo Horizonte:CNF
Belo Horizonte:PLU
Bemidji:BJI
Benbecula:BEB
Benghazi:BEN
Bengkulu:BKS

Screen shot 11



From: pr

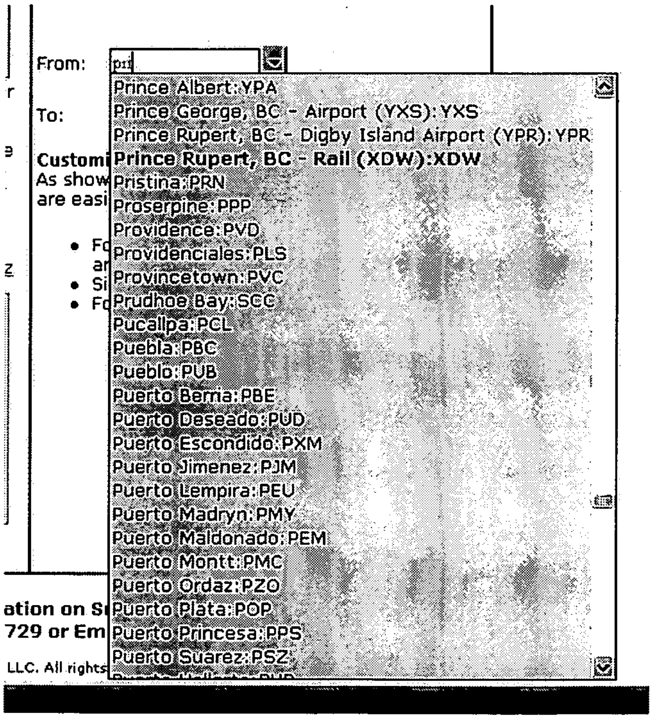
To:

Customi
As show
are easi

ation on S
729 or Em
LLC. All rights

Parana:PRA
Parent:XFE
Paris (all airports):PAR
Paris (Beauvais):BVA
Paris (CDG):CDG
Paris (Orly):ORY
Parkersburg:PKB
Parksville:XPB
Parma/Milan:PMF
Paro:PBH
Pasco:PSC
Pasni:PSI
Passo Fundo:PFB
Pasto:PSO
Patna:PAT
Pau:PUF
Paulatuk:YPC
Pavlodar:PWQ
Peace River:YPE
Peawanuck:YPO
Pedro Bay:PDB
Pekanbaru:PKU
Pelican:PEC
Pellston:PLN
Pemba, Mozambique (POL):POL

Screen shot 12



From: pri

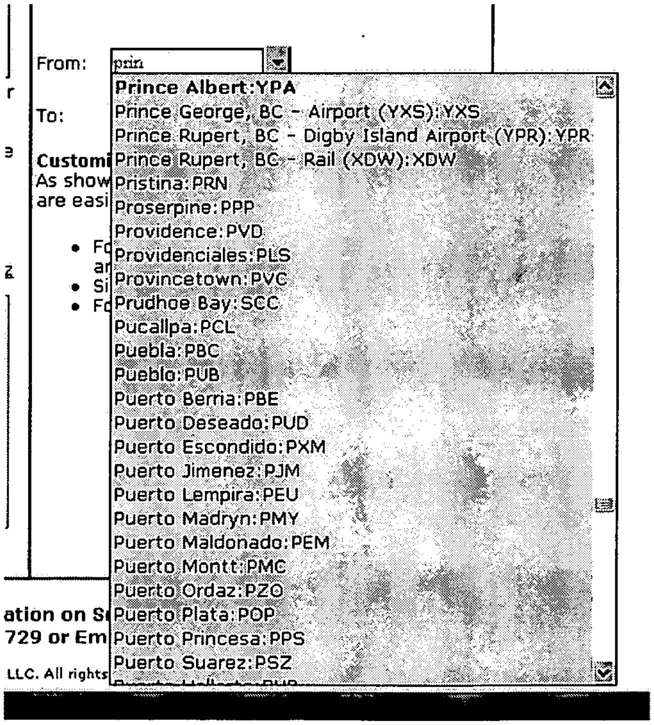
To:

Customi
As show
are easi

ation on S
729 or Em
LLC. All rights

Prince Albert:YPA
Prince George, BC - Airport (YXS):YXS
Prince Rupert, BC - Digby Island Airport (YPR):YPR
Prince Rupert, BC - Rail (XDW):XDW
Pristina:PRN
Proserpine:PPP
Providence:PVD
Providenciales:PLS
Provincetown:PVC
Prudhoe Bay:SCC
Pucallpa:PCL
Puebla:PBC
Pueblo:PUB
Puerto Berria:PBE
Puerto Deseado:PUD
Puerto Escondido:PXM
Puerto Jimenez:PJM
Puerto Lempira:PEU
Puerto Madryn:PMY
Puerto Maldonado:PEM
Puerto Montt:PMC
Puerto Ordaz:PZO
Puerto Plata:POP
Puerto Princesa:PPS
Puerto Suarez:PSZ

Screen shot 13



From: prin

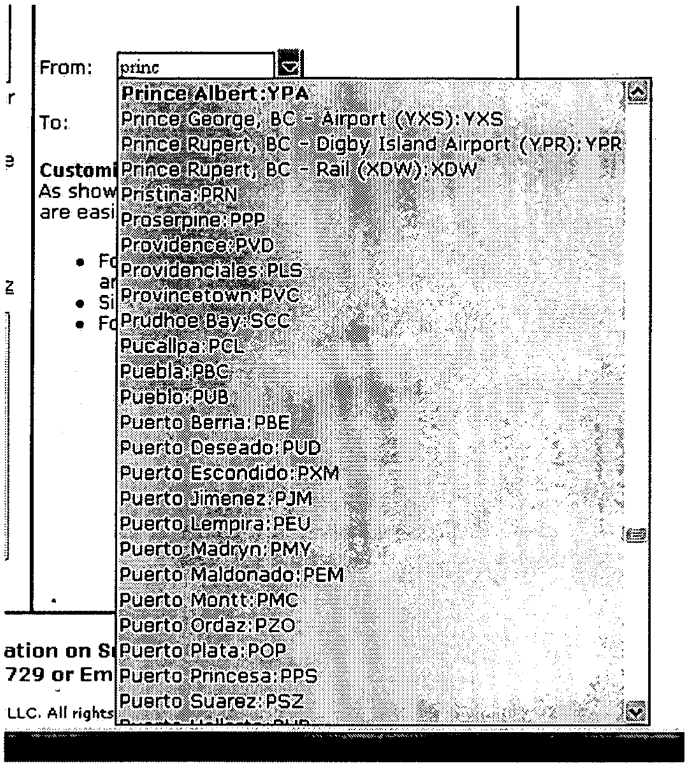
To:

Customi
As show
are easi

ation on S
729 or Em
LLC. All rights

Prince Albert:YPA
Prince George, BC - Airport (YXS):YXS
Prince Rupert, BC - Digby Island Airport (YPR):YPR
Prince Rupert, BC - Rail (XDW):XDW
Pristina:PRN
Proserpine:PPP
Providence:PVD
Providenciales:PLS
Provincetown:PVC
Prudhoe Bay:SCC
Pucallpa:PCL
Puebla:PBC
Pueblo:PUB
Puerto Berria:PBE
Puerto Deseado:PUD
Puerto Escondido:PXM
Puerto Jimenez:PJM
Puerto Lempira:PEU
Puerto Madryn:PMY
Puerto Maldonado:PEM
Puerto Montt:PMC
Puerto Ordaz:PZO
Puerto Plata:POP
Puerto Princesa:PPS
Puerto Suarez:PSZ

Screen shot 14



From: princ

To:

Customi

As show

are easi

- Fo ar
- Si
- Fo

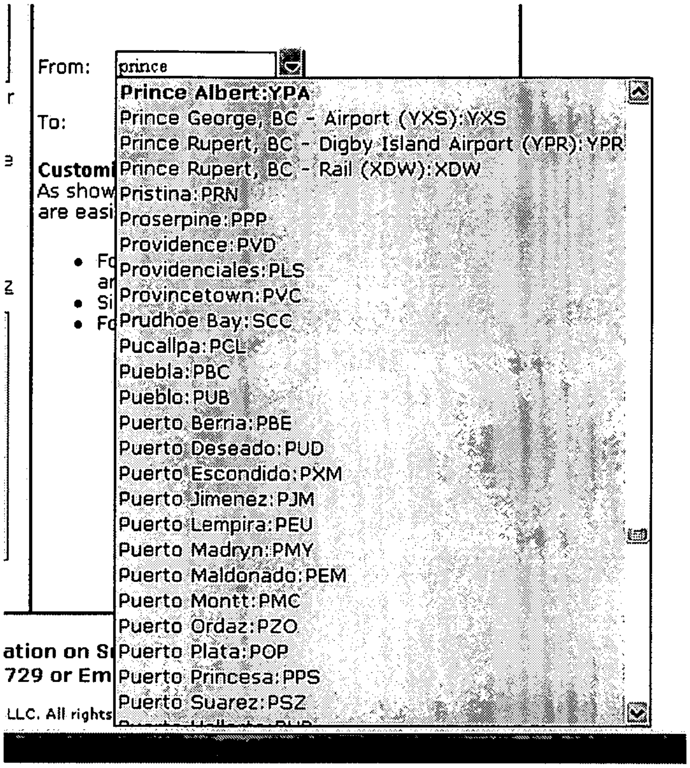
ation on S

729 or Em

LLC. All rights

Prince Albert:YPA
Prince George, BC - Airport (YXS):YXS
Prince Rupert, BC - Digby Island Airport (YPR):YPR
Prince Rupert, BC - Rail (XDW):XDW
Pristina:PRN
Proserpine:PPP
Providence:PVD
Providenciales:PLS
Provincetown:PVC
Prudhoe Bay:SCC
Pucallpa:PCL
Puebla:PBC
Pueblo:PUB
Puerto Berria:PBE
Puerto Deseado:PUD
Puerto Escondido:PXM
Puerto Jimenez:PJM
Puerto Lempira:PEU
Puerto Madryn:PMY
Puerto Maldonado:PEM
Puerto Montt:PMC
Puerto Ordaz:PZO
Puerto Plata:POP
Puerto Princesa:PPS
Puerto Suarez:PSZ

Screen shot 15



From: prince

To:

Customi

As show

are easi

- Fo ar
- Si
- Fo

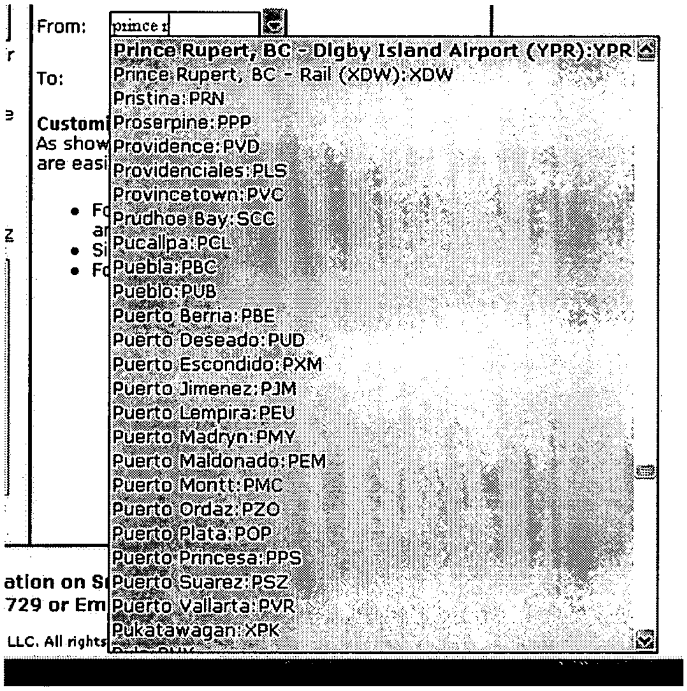
ation on S

729 or Em

LLC. All rights

Prince Albert:YPA
Prince George, BC - Airport (YXS):YXS
Prince Rupert, BC - Digby Island Airport (YPR):YPR
Prince Rupert, BC - Rail (XDW):XDW
Pristina:PRN
Proserpine:PPP
Providence:PVD
Providenciales:PLS
Provincetown:PVC
Prudhoe Bay:SCC
Pucallpa:PCL
Puebla:PBC
Pueblo:PUB
Puerto Berria:PBE
Puerto Deseado:PUD
Puerto Escondido:PXM
Puerto Jimenez:PJM
Puerto Lempira:PEU
Puerto Madryn:PMY
Puerto Maldonado:PEM
Puerto Montt:PMC
Puerto Ordaz:PZO
Puerto Plata:POP
Puerto Princesa:PPS
Puerto Suarez:PSZ

Screen shot 16



From: prince r

To:

Customi

As show

are easi

- Fo ar
- Si
- Fo

ation on S

729 or Em

LLC. All rights

Prince Rupert, BC - Digby Island Airport (YPR):YPR
Prince Rupert, BC - Rail (XDW):XDW
Pristina:PRN
Proserpine:PPP
Providence:PVD
Providenciales:PLS
Provincetown:PVC
Prudhoe Bay:SCC
Pucallpa:PCL
Puebla:PBC
Pueblo:PUB
Puerto Berria:PBE
Puerto Deseado:PUD
Puerto Escondido:PXM
Puerto Jimenez:PJM
Puerto Lempira:PEU
Puerto Madryn:PMY
Puerto Maldonado:PEM
Puerto Montt:PMC
Puerto Ordaz:PZO
Puerto Plata:POP
Puerto Princesa:PPS
Puerto Suarez:PSZ
Puerto Vallarta:PVR
Pukatawagan:XPK

Screen shot 17

Note that this is also the widest of the 3000 entries. As Screen shot 17 shows, the display field is just wide enough to allow full viewing of this entry, but not unnecessarily wide (with unneeded blank spaces to the right).

Claim 9 therefore defines a feature of the invention not shown in any of the cited prior art and provides advantages not now available to browser users. If in fact these features were as obvious as the Examiner asserts, then the applicants respectfully submit that they would be available, since browsers such as Microsoft IE have been around now for many years.

Claim 10 includes the claim limitations to both the "expandable display width" and the "multi-field searching" features. For reasons given above, each of these features is novel and non-obvious; claim 10 should therefore be allowable over the prior art, including the non-existent modified browser the Examiner hypothesizes in taking Official notice.

All three independent claims now include limitations that are not found or suggested by any of the prior art cited by the Examiner. Moreover, these features provide clear advantages over existing input methods, yet no one has yet implemented these features other than the inventors. The applicants therefore request that the claims all be allowed.

Date: 17 October 2005

Respectfully submitted,

Jeffrey Pearce

34825 Sultan-Startup Rd.
Sultan, WA 98294
Phone & fax: (360) 793-6687

Jeffrey Pearce
Reg. No. 34,729
Attorney for the Applicants